

What is claimed.

1. A programmable apparatus for automatic generation of a portable OOSQL query comprising:

5 a computer having a memory;

a program in said memory having,

a bridge between CORBA and EJB, and

a tie that delegates to said bridge and narrows one or more results to a desired EJB

object;

10 wherein the computer is directed by said software to allow EJB's to leverage CORBA  
functionality.

2. The bridge of claim 1 wherein said bridge further comprises:

a base class that calls into MOFW; and

15 a FH class containing OOSQL strings.

3. The method of claim 1 wherein said method is single.

4. The method of claim 1 wherein said method is extended.

20 5. The method of claim 1 wherein said method is lazy.

6. A method for generating a pre-deployment query for Enterprise Java Beans comprising the steps of:

verifying;

introspecting further comprising;

5 determining the methods of the specified input home interface;

ignoring all methods except those beginning with "find"

ignoring "findByPrimaryKey";

determining parameters expected by each qualifying method;

determining exceptions each qualifying method can raise;

10 determining return type of each qualifying method; and

emitting.

7. The verifying step of claim 6 further comprising the steps of:

parsing the input parameters;

15 verifying instantiation and inheritance from EJBHome;

determining an output file name;

8. The determining step of claim 7 further comprising:

starting with the input home interface name;

20 determining if a suffix "Home" exists;

responsive to determining that a suffix "Home" exists, removing said suffix "Home";

adding a suffix "Finder Helper"; and

determining a directory name for a file based on a package name.

9. The emitting step of claim 6 further comprising:

emitting (second) the source code for a Java class wherein said source code includes a  
5 package statement, appropriate import statements, the public class definition specifying the  
inheritance from FHB, the constructor, and each of the methods which qualified from the  
introspection step;

responsive to a method qualifying from the introspection step, emitting (third) a single  
line of code which has the method return the results of invoking one of the following methods,  
10 evaluate or singleEvaluate.

10. The emitting (third) step of claim 9 further comprising the step of switching from evaluate to  
extendedEvaluate, lazyEvaluate or extendedLazyEvaluate.

15 11. The emitting (third) step of claim 9 further comprising the step of switching from  
singleEvaluate to extendedSingleEvaluate.

12. Responsive to the invocation of a custom finder method, a method of using a FH class  
comprising:

delegating;

iterating (first)

adding;

SUB A17  
C/m 5  
12-2-1  
OK for  
7/17/10  
KChen

converting;

returning;

iterating (second);

narrowing

5

13. The delegating step of claim 12 wherein the delegating step commences when a custom finder method is invoked and the Tie delegates to the finder helper class.

10

14. The invoking step of claim 12 wherein said step involves making use of the query capabilities of the underlying application server.

15

15. The invoking step of claim 12 wherein said step involves CB and involves calling the evaluate( ) or extendedEvaluate ( ) methods of the IQueryableIterableHome class.

15

16. The Iterating (first) step of claim 12 wherein said step involves accessing each object in the set returned from the invoking step.

17. The adding step of claim 12 wherein said step involves keeping track of each object accessed in the invoking step by temporarily storing each of them in a Vector.

20

18. The converting step of claim 12 wherein said step uses a method on the Vector class called "elements ( )", which creates an Enumeration out of the Vector that was built up in the adding

step.

19. The returning step of claim 12 wherein said step involves the finder helper returning the Enumeration from the converting step to the Tie.

5

20. The Iterating (second) step of claim 12 wherein said step is an iteration over the portable Enumeration class returned by the finder helper.

21. The Narrowing step of claim 12 wherein said step involves using the standard PortableRemoteObject.narrow(), whereby application server specific code is avoided by staying with portable code after return from the call to the finder helper class.

22. A computer readable memory for causing a computer to allow EJBs to leverage CORBA functionality comprising:

15 a computer readable storage medium;

a computer program stored in said storage medium;

wherein the storage medium, so configured by said computer program, causes the computer to bridge between CORBA and EJB, to provide a Finder Helper Base class which encapsulates application server-specific logic, and to generate a finder helper subclass that contains portable OOSQL strings.

20

23. The computer program of claim 22 further comprising:

a bridge between CORBA and EJB comprising a base class that calls into MOFW and a  
FH class containing OOSQL strings; and

a tie that delegates to said bridge and narrows one or more results to a desired EJB object;

5

10

15

20

25